



# TECNOLOGIA EM DESENVOLVIMENTO DE SOFTWARE MULTIPLATAFORMA

TÉCNICA DE PROGRAMAÇÃO I



TÉCNICA DE PROGRAMAÇÃO I

PROFº LUIZ CLÁUDIO

# PROJETO CRUD E TELA MENU

## PROJETO CRUD USUÁRIO

Código:

Login:

Senha:

Telefone: ( ) -



Cadastrar



Alterar



Excluir

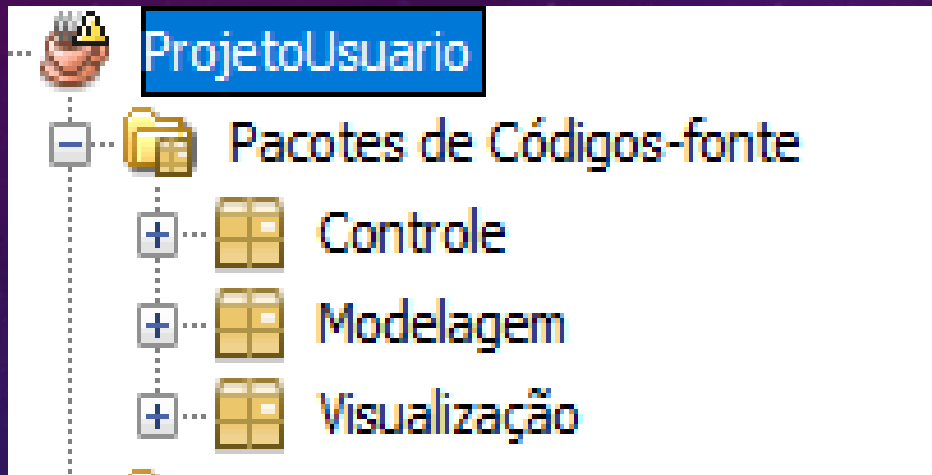


Limpar

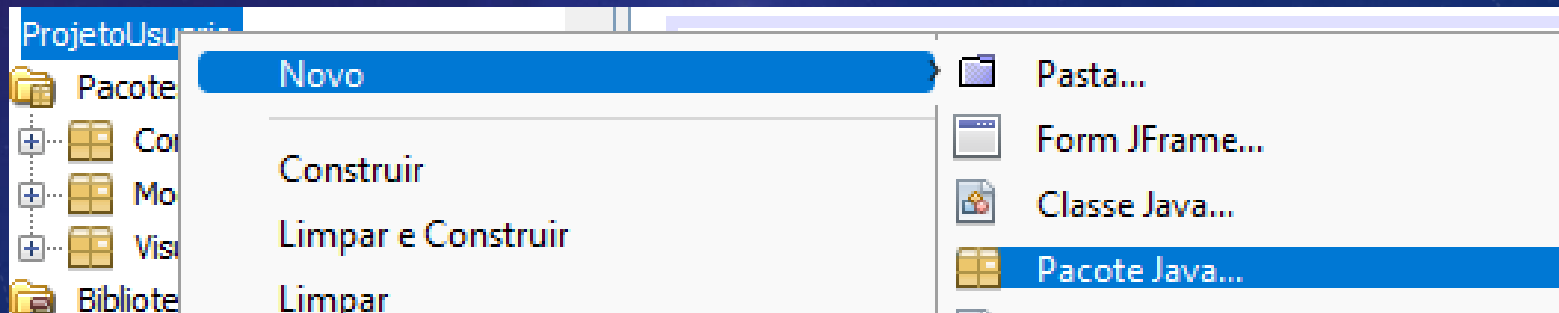
Código	Login	Senha	Telefone



# Crie o Projeto do Usuário e os pacotes



Clique com o botão direito no nome do Projeto e crie os Pacotes MVC – Modelagem, Controle e Visualização



# Crie classe Conexao-pacote Controle

```
package Controle;

import java.sql.*;
import javax.swing.JOptionPane;

public class Conexao {

    final private String driver = "com.mysql.jdbc.Driver";
    final private String url= "jdbc:mysql://127.0.0.1/banco";

    final private String usuario="root";
    final private String senha="";
    private Connection conexao;// objeto que faz conexao com o banco
    public Statement statement;// objeto que abre caminho até o banco, cria a autoestrada.
    public ResultSet resultSet;// objeto que armazena os comandos sql

    public boolean conecta() {
        boolean result = true;

        try {
            Class.forName(driver);
            conexao = DriverManager.getConnection(url,usuario,senha);
            //JOptionPane.showMessageDialog(null,"Conectou com o Banco de Dados");

        } catch (ClassNotFoundException Driver){
            JOptionPane.showMessageDialog(null,"Driver nao localizado: "+Driver);
            result = false;
        } catch (SQLException Fonte) {
            JOptionPane.showMessageDialog(null,"Erro na conexão com a fonte de dados: "+Fonte);
            result = false;
        }
        return result;
    }
}
```

Endereço do mysql  
Pode usar alguns casos localhost

# Classe Conexao

```
public void desconecta () {
    boolean result = true;
    try
    {
        conexao.close();
        //JOptionPane.showMessageDialog(null, "Banco fechado");
    }
    catch(SQLException fecha)
    {
        JOptionPane.showMessageDialog(null, "Não foi possível fechar o banco de dados" + fecha);
        result = false;
    }
}

public void executeSQL(String sql) {
//chamada do metodo conecta para abrir a conexão com o db
    conecta();
    try{

        statement = conexao.createStatement();

        statement.execute(sql);
        //desconecta();
    }catch(SQLException sqle){
        JOptionPane.showMessageDialog(null, "Driver não encontrado!" + sqle.getMessage());
    }
}
```

Crie a classe Usuario no pacote de Modelagem  
Crie os atributos.

```
public class Usuario {  
    private int codigo;  
    private String login;  
    private String senha;  
    private String Telefone;  
    //IMPORTE A CLASSE DE CONEXÃO  
    Conexao con= new Conexao();  
    //FAÇA O CONSTRUTOR  
    public Usuario() {  
        this(0,"","","");  
    }  
    public Usuario(int codigo, String login, String senha,  
        this.codigo = codigo;  
        this.login = login;  
        this.senha = senha;  
        this.Telefone = Telefone;  
    }  
    //FAÇA O ENCAPSULAMENTO  
    public int getCodigo() {  
        return codigo;  
    }  
    public void setCodigo(int codigo) {  
        this.codigo = codigo;  
    }  
    public String getLogin() {  
        return login;  
    }  
}
```

## Usuario

-codigo: int  
-login: String  
-senha: String  
-telefone: String

+cadastrarUsuario()  
+consultar()  
+excluir()  
+alterar()



# Crie os métodos cadastrar, alterar e excluir

```
public void cadastrarUsuario(){
    String sql= "Insert into usuario(Codigo,Telefone,Login,Senha)values "+
        "("+ this.getCodigo()+"," +this.getTelefone()+"," +this.getLogin()+"," + this.getSenha()+"" )";
    con.executeSQL(sql);
    JOptionPane.showMessageDialog(null, "Registrado c sucesso");
}

public void excluir(){
    String sql;
    sql= "Delete from usuario where codigo="+ getCodigo();
    con.executeSQL(sql);
    JOptionPane.showMessageDialog(null, "Registro excluido com sucesso...");
}

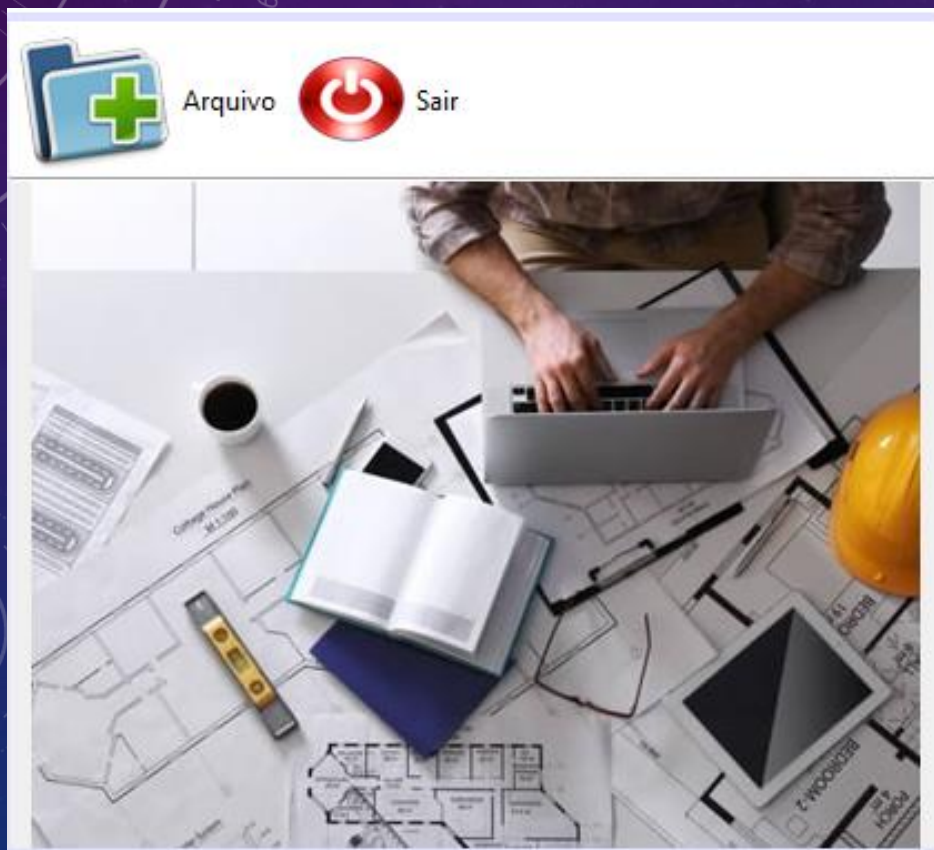
public void alterar(){
    String sql;
    sql= "Update usuario set  login='"+this.getLogin()+"' , senha= '" + this.getSenha()+ "' , "
        + "telefone= '" + this.getTelefone()+ "' where codigo="+ this.getCodigo();
    con.executeSQL(sql);
    JOptionPane.showMessageDialog(null, "Registro alterado com sucesso...");
}
```

# O método consultar

```
public ResultSet consultar() {  
    ResultSet tabela;  
    tabela = null;  
    String sql= "Select * from usuario";  
    tabela = con.RetornarResultset(sql);  
    return tabela;  
}
```



No pacote Visualização. Crie o Formulário FMenu dentro do pacote Visualização



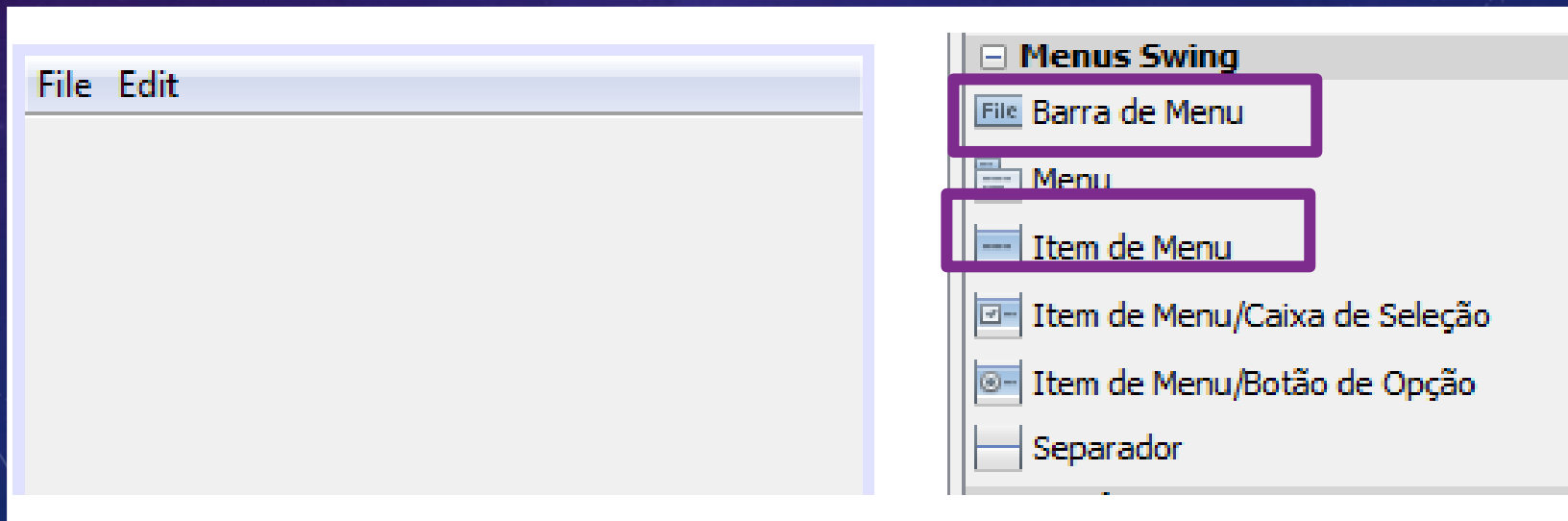


**NetBeans**

CRIAÇÃO DE MENUS

# Criando Menu

- Adicione um novo JFrame ao projeto.
- Acesse a "**Paleta**" do Netbeans, na categoria "**Menus Swing**" clique em "**Barra de Menu** "(JMenuBar1) e arraste até o JFrame, em seguida acrescente um Item de menu conforme figura a seguir



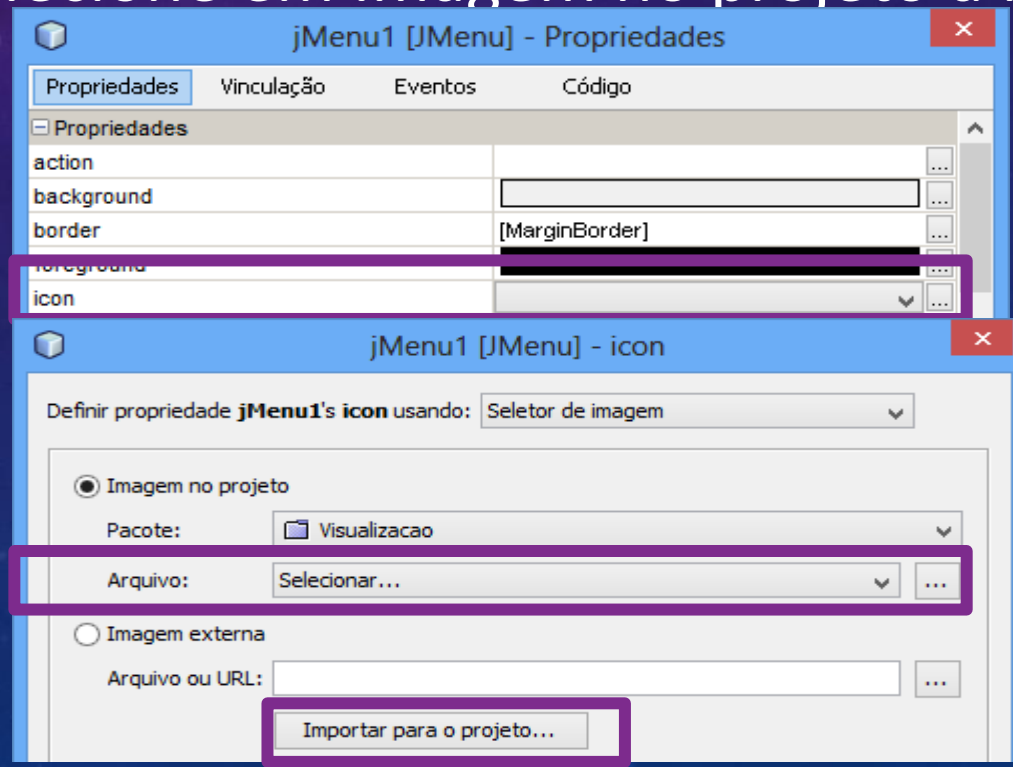
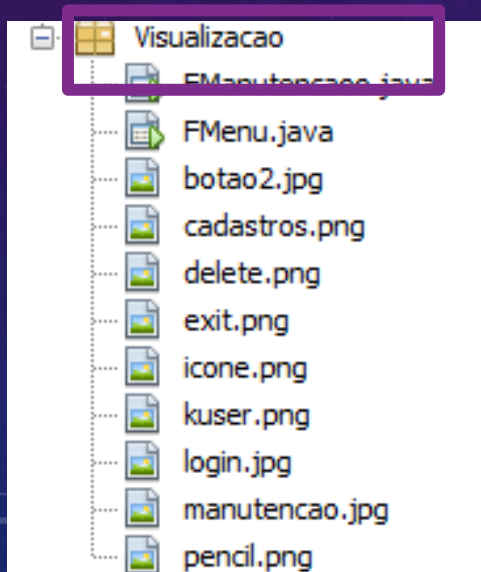
# Criando Menu

- Por padrão o JMenuBar1 já vem com dois menus (File e Edit). Se você clicar em cima dos Menus verá que o nome deles são respectivamente: JMenu1, JMenu2. Só a barra de menu (JMenuBar) com os 2 menus(JMenu) não fica legal. Para que um menu fique funcional, necessitamos inserir os Itens de Menu.



# Criando Menu- adicionar ícones

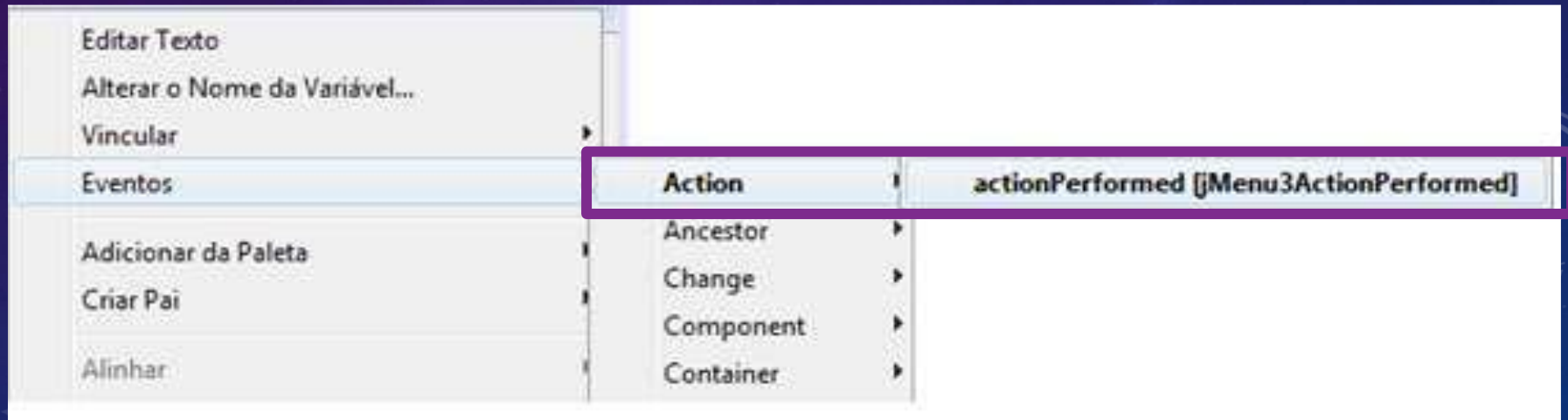
- Clique Na propriedade icon... Clique no botão importar para o projeto
- Ou para facilitar arraste as imagens para o pacote Visualização e selecione em Imagem no projeto a imagem





# Criando Menu- chamar outro formulário menu

- Clique com o botão direito em cima do Item de Menu que vai chamar o formulário e acesse o evento **actionPerformed**.





# Criando Menu- chamar outro formulário menu

- Para abrir uma janela é necessário instanciar uma instancia da classe do formulário e
- enviar **true** no atributo visible com o seguinte código:

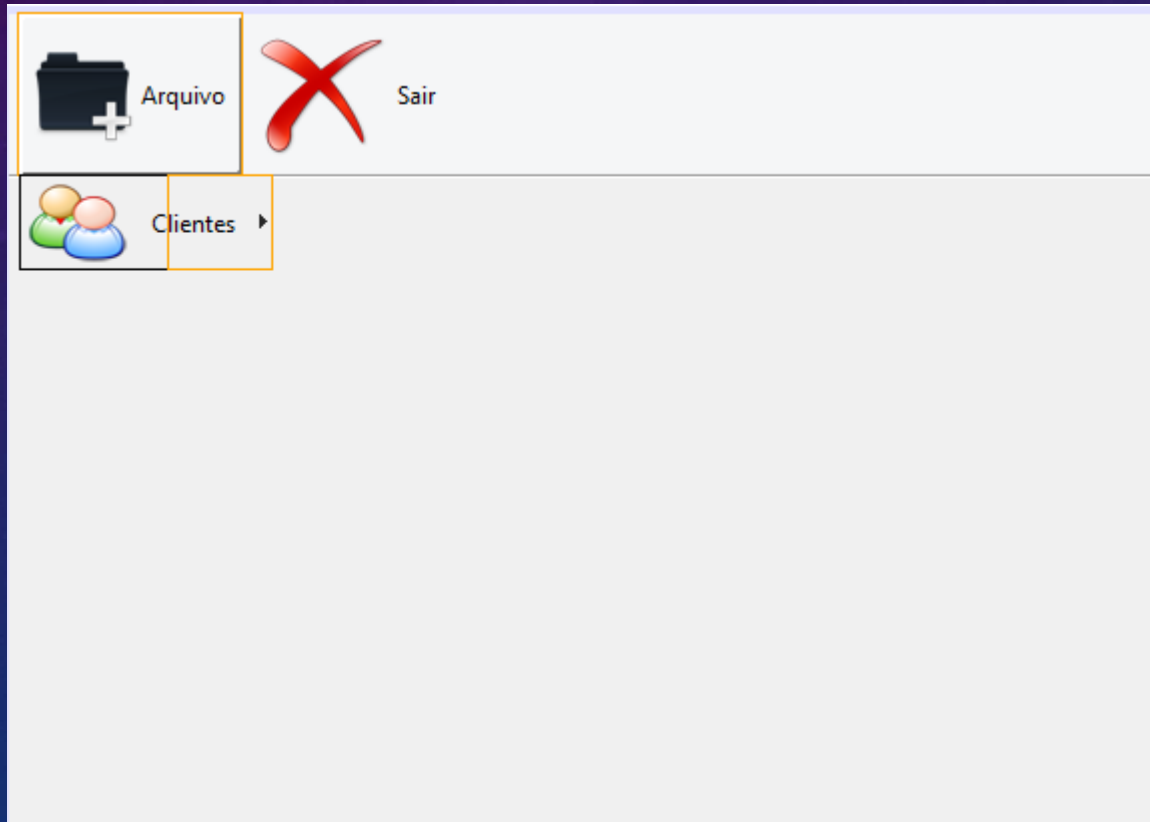
```
NomeDoFormulario NomeDoObjetoDeFormulario = new NomeDoFormulario();  
NomeDoObjetoDeFormulario.setVisible(true);
```

EXEMPLO:

```
FUusuario fu = new FUusuario();  
fu.setVisible(true);
```

# Criando Menu- chamar outro formulário menu

- Crie o menu abaixo:



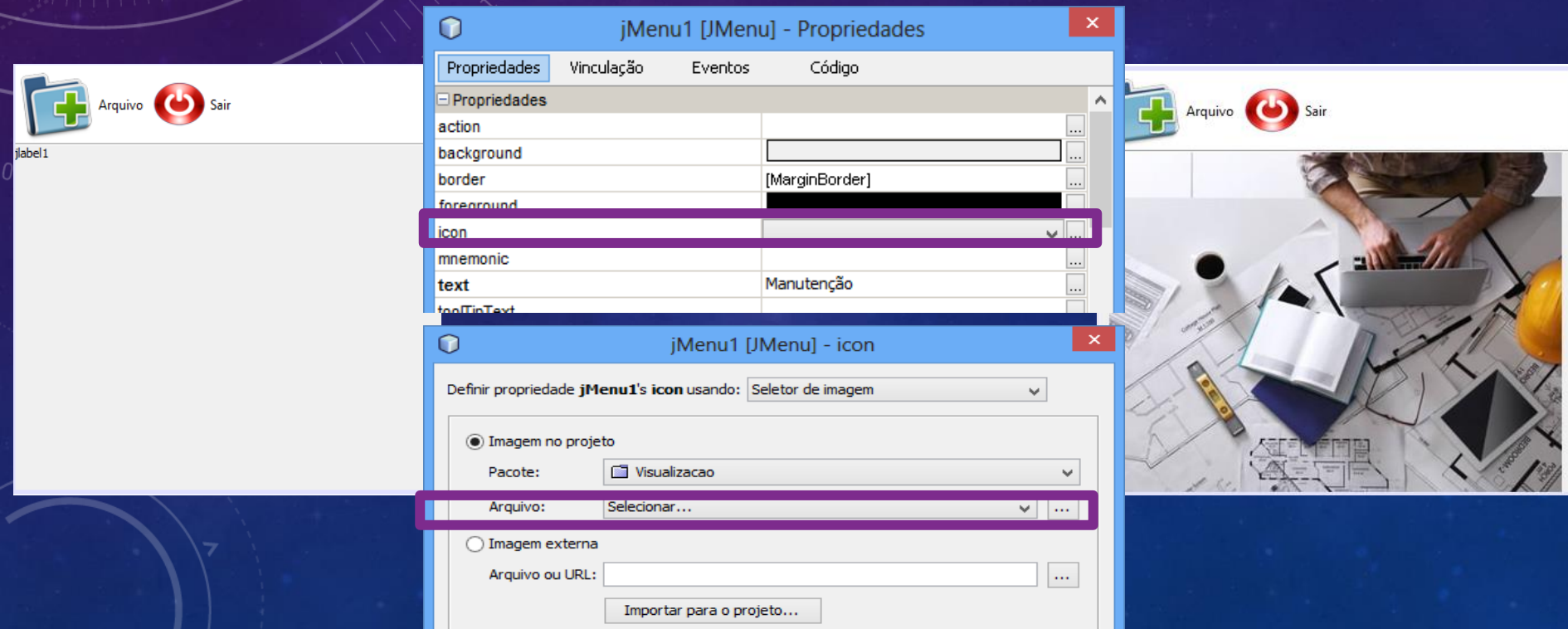
# Criando Menu- chamar outro formulário menu

- Para que o formulário que foi aberto não encerre a aplicação quando ele for fechado configure no formulário que foi chamado a propriedade `defaultCloseOperation` com:

<code>defaultCloseOperation</code>	<code>DISPOSE</code>	▼	⌵
------------------------------------	----------------------	---	---

# Para adicionar uma Imagem ao Formulário

ADICIONE UM LABEL E ALTERE A PROPRIEDADE ICON E SELECIONE A IMAGEM DESEJADA EM IMAGEM DO PROJETO OU IMPORTE A IMAGEM :



No pacote Visualização. Crie o Formulário FUsuario dentro do pacote Visualização

## PROJETO CRUD USUÁRIO

Código:

Componente JTextField /campo de texto  
Nome da variável: txtCodigo

Login:

Componente JTextField/ campo de texto  
Nome da variável: txtLogin

Senha:

Componente JPasswordField/ campo de senha  
Nome da variável: txtSenha

Telefone:

Componente JFormattedTextField/ campo formatado  
Nome da variável: txttelefone



Cadastrar



Alterar



Excluir



Limpar

Código	Login	Senha	Telefone

# Alterando a máscara do campo formatado

## PROJETO CRUD US

Código:

Login:

Senha:

Telefone: ( ) -

txtTelefone [JFormattedTextField] - Propriedades			
Propriedades	Vinculação	Eventos	Código
border		[XPFillBorder]	...
columns		0	...
document		<default>	...
foreground		■ [0,0,0]	...
formatterFactory		[AbstractFormatterFactory]	...
horizontalAlignment		LEADING	...
text		( ) -	...

txtTelefone [JFormattedTextField] - formatterFactory

Definir propriedade txtTelefone's formatterFactory usando: Editor de Formato

Categoria	Formatar	Formato:
número	###-###	
data	personalizado	(##)#####-###
hora		
porcentagem		
moeda		
máscara		

Exemplo

Valor a ser formatado:  Teste

Resultado:

Para colocar uma máscara para o telefone selecione a propriedade formatterFactory,

Selecione mascara->personalizado e defina o formato da máscara



No Formulário FUsuario dentro do pacote Visualização , crie os botões e coloque ícones alterando a propriedade icon

## PROJETO CRUD USUÁRIO

Código:

Login:

Senha:

Componente JButton  
Nome da variável: btnCadastrar

Componente JButton  
Nome da variável: btnAlterar

Componente JButton  
Nome da variável: btnExcluir



Cadastrar



Alterar



Excluir



Limpar

Código	Login	Senha	Telefone

Componente JButton  
Nome da variável: btnLimpar

No Formulário FUsuario dentro do pacote Visualização ,  
coloque a tabela de visualização dos dados

**PROJETO CRUD USUÁRIO**

**tblUsuarios [JTable] - Propriedades**

Propriedades	Vinculação	Eventos	Código
background		<input type="checkbox"/> [255,255,255]	...
border		(Sem Borda)	...
font		Tahoma 11 Simples	...
foreground		■ [0,0,0]	...
<b>model</b>		[Modelo de Tabela]	...

Componente Jtabela / Tabela  
Nome da variável: tblUsuarios

Para adicionar as colunas na tabela altere a propriedade **model** e defina os títulos das colunas

Modelo de Tabela

Definições da Tabela | Valores Default

Especifique os Tipos de Título e Coluna Aqui:

Coluna	Título	Tipo	Editável
1	Código	Object	<input checked="" type="checkbox"/>
2	Login	Object	<input checked="" type="checkbox"/>
3	Senha	Object	<input checked="" type="checkbox"/>
4	Telefone	Object	<input checked="" type="checkbox"/>

Botões: Inserir, Excluir, Mover para Cima, Mover para Baixo

No Formulário Fusuario , no inicio do código instancie a classe de Conexao e Usuario

```
package Visualização;

import Controle.Conexao;
import Modelagem.Usuario;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author convidado_
 */
public class FUsuario extends javax.swing.JFrame {

    Usuario usu = new Usuario();
    Conexao con = new Conexao();
}
```

# Código Botão Cadastrar

```
private void btnCadastrarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:
```

```
    usu.setCodigo(Integer.parseInt(txtCodigo.getText()));  
    usu.setTelefone(txtTelefone.getText());  
    usu.setSenha(txtSenha.getText());  
    usu.setLogin(txtLogin.getText());  
    usu.cadastrarUsuario();
```

Pega as informações digitadas nas cixas de texto envia aos atributos da classe usuário

```
    ResultSet tabela;  
    tabela = null;
```

Faz a consulta dos dados no banco e retorna na tabela tblUsuarios

```
    tabela = usu.consultar();  
    DefaultTableModel modelo = (DefaultTableModel) tblUsuarios.getModel();  
    modelo.setNumRows(0);
```

```
    try  
    {  
        do{  
            modelo.addRow(new String[]{tabela.getString(2), tabela.getString(3), tabela.getString(4), tabela.getString(5)});  
        }  
        while(tabela.next());  
    }catch(SQLException erro)  
    {  
        JOptionPane.showMessageDialog(null, "Erro ao preencher tabela"+ erro) ;  
    }  
}
```

Adiciona os dados do banco nas colunas da tabela, os números do getString deve ser alterado caso não mostre na posição correta o dado na tabela

# Os botões Excluir e Alterar

```
private void btnAlterarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    usu.setCodigo(Integer.parseInt(txtCodigo.getText()));  
    usu.setLogin(txtLogin.getText());  
    usu.setSenha(txtSenha.getText());  
    usu.setTelefone(txtTelefone.getText());  
    usu.alterar();  
}
```

Pega as informações digitadas nas cixas de texto  
envia aos atributos da classe usuário, e chama o  
método alterar

```
private void btnExcluirActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    usu.setCodigo(Integer.parseInt(txtCodigo.getText()));  
    usu.excluir();  
}
```

Pega o código a ser excluído e envia a classe  
usuário, e chama o método excluir

# O botão Limpar

```
private void btnLimparActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    txtCodigo.setText("");  
    txtLogin.setText("");  
    txtSenha.setText("");  
    txtTelefone.setText("");  
}
```



# O botão Sair Formulário Menu



Clique com botão direito no botão sair Eventos  
Action ActionPerformed e código:

```
private void jMenuItem2ActionPerformed(  
    // TODO add your handling code here:  
    System.exit(0);  
}
```

